REMARKS

In response to the Office Action mailed February 13, 2009, Applicants respectfully request reconsideration. Claims 1-15 were previously pending in this application. By this amendment, claims 1, 7, 8 and 11-15 have been amended. No new claims have been added. As a result, claims 1-15 are pending for examination with claims 1, 7, 8 and 11 being independent. No new matter has been added.

Objections to the Claims

The Office Action objected to claim 1 because of an informality, an inadvertent duplication of the word "that." Applicants have herein deleted the occurrence of the second "that."

Accordingly, withdrawal of this objection is respectfully requested.

Rejections under 35 U.S.C. §112

The Office Action rejected claims 12-15 under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicants regard as the invention. Applicants have amended claims 12-15 to delete a limitation stating that "a modification ... is reduced."

Accordingly, withdrawal of this rejection is respectfully requested.

Rejections Under 35 U.S.C. §103

The Office Action rejected claims 1-15 under 35 U.S.C. §103(a) as purportedly being unpatentable over Nexus 5001 Forum: Standard for a Global Embedded Processor Debug Interface (Nexus 5001 Forum), hereinafter "Nexus," in view of Argade et al., U.S. Patent No. 5,724,505, hereinafter "Argade." Applicants respectfully disagree.

A. Independent Claim 1

Claim 1 has been amended to recite, *inter alia*,

> ***after*** it is determined that the address of the destination instruction is not explicitly indicated in the jump instruction:

> assigning a second value to the first set of bits of at least one digital message to provide an implicit jump message indicating an occurrence of an implicit jump,
>
> adding a field to the implicit jump message, the field comprising a second set of bits identifying a type of the implicit jump from among several types of implicit jumps, wherein the field is added **after** it is determined that the address of the destination instruction is not explicitly indicated in the jump instruction, and
>
> transmitting the implicit jump message (emphasis added).

On page 5, the Office Action concedes that Nexus "does not explicitly disclose adding a field to the implicit jump message, the field comprising a second set of bits identifying a type of the implicit jump from among several types of implicit jumps, wherein the field is added only when the address of the destination instruction is not explicitly indicated in the jump instructions." As mentioned above, claim 1 has been amended to recite that *the field is added after it is determined that the address of the destination instruction is not explicitly indicated in the jump instruction* (emphasis added).

Applicants respectfully state that Argade does not teach or suggest adding a field to the implicit jump message, the field comprising a second set of bits identifying a type of the implicit jump from among several types of implicit jumps, wherein *the field is added after it is determined that the address of the destination instruction is not explicitly indicated in the jump instruction*, as recited in claim 1 (emphasis added). Indeed, Argade states that INSTR_TYPE signals provided over an instruction-type line. Argade discusses that, by identifying an INSTR_TYPE as one of the three types of pre-defined discontinuities, the TBC block 50 may determine *whether to record or to discard its corresponding address* (or addresses) and *whether additional information about the INSTR_TYPE needs to be recorded* (Argade, col. 5, lines 43-48). Thus, in Argade, **first**, INSTR_TYPE signals for the three types of discontinuities are send, after which they are used to determine whether to record or discard a corresponding address.

In Argade, the TBC block 50 acquires the INSTR_TYPE signal from the processor core 12 via, the line 30, and then acquires the DEST_ADDR of that INSTR_TYPE form the arbitrator block 22 via the inter-module bus 28 (Argade, col. 7, lines 20-24). Argade states that, at test 112, the TBC block 50 determines **whether the INSTR_TYPE** received at step 108 **is of type_1** and, if it is, the TBC block 50 acquires the corresponding RETURN_ADDR from the arbitrator block 22 at step 114 (Argade, Fig. 3; col. 7, lines 24-27) (emphasis added). Further, Argade

states that, at test 132, the TBC block 50 **determines whether the INSTR_TYPE** received at step 108 **is of type_2**, and, if it is, at test 134 the TBC block 50 checks if the address FIFO 52 is full ... (Argade, Fig. 4; col. 7, lines 46-49) (emphasis added). Further, if at test 132 the TBC block 50 determined that the INSTR_TYPE was not type_2, at test 146 the TBC block 50 **determines if the INSTR_TYPE is type_3** and, if it is, the TBC block 50 discards the DEST_ ADDR at step 148 (Argade, Fig. 4; col. 7, lines 63-66) (emphasis added). Thus, Argade clearly uses the INSTR_TYPE signal to determine whether to record or discard address information. Accordingly, Argade does not teach or suggest *after it is determined that the address of the destination instruction is not explicitly indicated in the jump instruction*: assigning a second value to the first set of bits of at least one digital message to provide an implicit jump message indicating an occurrence of an implicit jump, *adding a field to the implicit jump message*, the field comprising a second set of bits identifying a type of the implicit jump from among several types of implicit jumps, wherein the field is added after it is determined that the address of the destination instruction is not explicitly indicated in the jump instruction, and transmitting the implicit jump message, as recited in claim 1 (emphasis added).

On page 6, the Office Action states that Argade teaches "it would have been further obvious to one or ordinary skill in the art at the time of the invention to include the trace information of Argade into the digital messages of Nexus in order to decrease cost and space, for example (e.g. incorporating the type of implicit jump in the tracing mechanisms already existing in Nexus, as opposed to creating some entirely new method of trace information transmission solely for the use of conveying implicit jump type information, would obviate the cost of this hypothetical new method of trace information transmission as well as the space necessary to implement this hypothetical new method of trace information transmission. The implementation of the trace information of Argade into the digital messages of Nexus thus teaches the limitations of adding a field to the digital message, the field comprising a second set of bits identifying a type of the implicit jump from among several types of implicit jumps."

Applicants respectfully assert that Nexus, even if combined with Argade, does not teach limitations of claim 1. Argade discloses three types of the INSTR_TYPE signals. Each of the three types of signals is necessary for operation of the system of Argade. Indeed, as discussed above, the TBC block 50 of Argade **determines if the INSTR_TYPE is type_3** and, **if it is, the TBC block 50 discards the DEST_ ADDR** at step 148 (Argade, Fig. 4; col. 7, lines 64-66)

(emphasis added). Further, in another portion, Argade states that if at test 240 the TBC block 50 determined that the INSTR_TYPE was not of type_3, at step 246 the TBC block 50 shifts the OVERFLOW code from the INSTR_TYPE FIFO 54 to the JTAG port 44 to be transmitted to the external debug host computer 100 (Argade, Fig. 7; col. 9, lines 41-46). Therefore, for proper operation of Argade, all three types of the INSTR_TYPE signals are required to be sent. Indeed, as discussed above, the TBC block 50 of Argade determines **if the INSTR_TYPE is type_3** (emphasis added).

Furthermore, as discussed above, in Argade, the TBC block 50 acquires the INSTR_TYPE signal from the processor core 12 *via, the line 30,* and then acquires the DEST_ADDR of that INSTR_TYPE form the arbitrator block 22 *via the inter-module bus 28* (Argade, col. 7, lines 20-24) (emphasis added). Therefore, the line 30 is used to provide the INSTR_TYPE signal to the TBC block 50 and a different source, the inter-module bus 28 is used to provide the DEST_ADDR of that INSTR_TYPE to the TBC block 50. One of skill in the art would understand that this is different from "after it is determined that the address of the destination instruction is not explicitly indicated in the jump instruction, ... adding a field to the implicit jump message, the field comprising a second set of bits identifying a type of the implicit jump from among several types of implicit jumps," as recited in claim 1.

The Office Action states that "the implementation of the trace information or Argade into the digital messages of Nexus thus teaches the limitations of adding a field to the digital message, the field comprising a second set of bits identifying a type of the implicit jump from among several types of implicit jumps." However, Argade discusses INSTR_TYPE *signals* provided **separately** from the DEST_ADDR of each of the INSTR_TYPE signals. Furthermore, Argade discusses that, when the INSTR_TYPE is of type_1, at step 126, the TBC block 50 *stores the INSTR_TYPE* acquired at step 108 *in the INSTR_TYPE FIFO 54* (Argade, col. 7, lines 37-39) (emphasis added). At step 128 of Argade, the TBC block 50 *stores the RETURN_ADDR and DEST_VECTOR in the address FIFO 52* (Argade, col. 7, lines 42-43) (emphasis added). Similarly, when the INSTR_TYPE is of type_2, the TBC block 50 *stores the INSTR_TYPE* acquired at step 108 *in the INSTR_TYPE FIFO 54* and *the DEST_ADDR in the address FIFO 52* (Argade, col. 7, lines 55-60) (emphasis added). Thus, not only the INSTR_TYPE signals are sent to the TCB block separately from the corresponding addresses, but they are also stored in a separate FIFO, which is again different from adding a field to an implicit jump message as

recited in claim 1. Combining the outputs of the address FIFO 52 and INSTR_TYPE FIFO 54 into a single signal by a multiplexer 56 (Argade, col. 6, lines 23-27) is also different from limitations of claim 1.

Further, **each** executed instruction has an INSTR_TYPR signal associated with it (Argade, col. 5, lines 39-41) (emphasis added). There is no suggestion in Argade that it may be appropriate to use only the type_1 and type_2 types of discontinuities and not the type_3.

The Office Action states that it would have been further obvious to one of ordinary skill in the art at the time of the invention to include the trace information of Argade, which identifies types of implicit jumps, *in only the implicit jump messages of Nexus*, in order to increase efficiency, *as an explicit jump message of Nexus would logically have no use for information identifying a type of implicit jump (as an explicit jump is not an implicit jump at all),* and forgoing the field in explicit jump messages would decrease the size of the explicit jump messages, thus increasing transmission efficiency (emphasis added). Applicants respectfully state that none of the cited references teaches or suggests "adding a field to the implicit jump message, the field comprising a second set of bits identifying a type of the implicit jump from among several types of implicit jumps," as recited in claim 1. Further, none of the references suggests that this would "increase efficiency." As discussed above, Argade clearly states that all three of the INSTR_TYPE signals, inducing the type_3 which, as the Office Action appears to state, has "no use" in Nexus, are required. The TCB block of Argade determines whether the INSTR_TYPE is of type_3. Thus, the type_3 is required in Argade.

The Office Action states that an explicit jump message of Nexus would logically have no use for information identifying a type of implicit jump (as an explicit jump is not an implicit jump at all), and forgoing the field in explicit jump messages would decrease the size of the explicit jump messages, thus increasing transmission efficiency. However, it is not clear why, if one were to combine Nexus and Argade, as suggested in the Office Action, one would not provide a signal, such as the type_3 of Argade, for the direct branch message of Nexus. Thus, Argade does not cure the deficiency of Nexus.

Accordingly, claim 1 patentably distinguishes over Nexus and Argade, either alone or in combination.

Claims 2-6 and 12 depend from claim 1 and are allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 1-6 and 12 is respectfully requested.

B. Independent Claim 7

Claim 7 has been amended to recite, *inter alia*, that **after it is determined that the address of the destination instruction is not explicitly indicated in the jump instruction and the first set of bits is set to the second value**, the generation means adds a field to the implicit jump message, the field comprising a second set of bits identifying a type of the implicit jump from among several implicit jump types (emphasis added). As discussed above, neither Nexus nor Argade teaches or suggests this limitation.

Accordingly, claim 7 patentably distinguishes over Nexus and Argade, either alone or in combination.

Claim 13 depends from claim 7 and is allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 7 and 13 is respectfully requested.

C. Independent Claim 8

Claim 8 has been amended to recite, *inter alia*,

> determining whether the jump is associated with a jump instruction explicitly indicating an address of the jump destination instruction;
>
> generating at least one digital message upon the detection of the jump, wherein
>
> **only after it is determined that the jump is associated with a jump instruction not explicitly indicating the address of the jump destination instruction**:
>
> determining that the jump is implicit,
>
> generating the at least one digital message as an implicit jump message indicating an occurrence of the implicit jump, and
>
> adding an additional field to the implicit jump message, wherein the additional field includes a value identifying a type of the implicit jump (emphasis added).

As discussed above, neither Nexus nor Argade teaches or suggests this limitation.

Accordingly, claim 8 patentably distinguishes over Nexus and Argade, either alone or in combination.

Claims 9, 10 and 14 depend from claim 8 and are allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 8-10 and 14 is respectfully requested.

D. Independent Claim 11

Claim 11 has been amended to recite, *inter alia*, that **after it is determined that the jump is associated with a jump instruction not explicitly indicating an address of the jump destination instruction**, adding a field to at least one digital message to provide the at least one digital message as an implicit jump message transmitted on the execution of the instruction sequence by the microprocessor and indicating an occurrence of an implicit jump, wherein the field includes a value identifying a type of the implicit jump (emphasis added). As discussed above, neither Nexus nor Argade teaches or suggests this limitation.

Accordingly, claim 11 patentably distinguishes over Nexus and Argade, either alone or in combination.

Claims 9, 10 and 14 depend from claim 8 and are allowable for at least the same reasons.

Claim 15 depends from claim 11 and is allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 11 and 15 is respectfully requested.

CONCLUSION

A Notice of Allowance is respectfully requested. The Examiner is requested to call the undersigned at the telephone number listed below if this communication does not place the case in condition for allowance.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicants hereby request any necessary extension of time. Applicants believe no fee is due with this response. However, if a fee is due, please charge our Deposit Account No. 23/2825 under Docket No. S1022.81242US00 from which the undersigned is authorized to draw.

Dated: May 13, 2009

Respectfully submitted,

By: _____

James H. Morris, Reg. 34,681
WOLF, GREENFIELD & SACKS, P.C.
Federal Reserve Plaza
600 Atlantic Avenue
Boston, Massachusetts 02210-2206
617.646.8000